1. What is the IP address used by your computer (**source**)?

192.168.0.9

2. On what port number is your computer sending and receiving TCP segments for this connection?

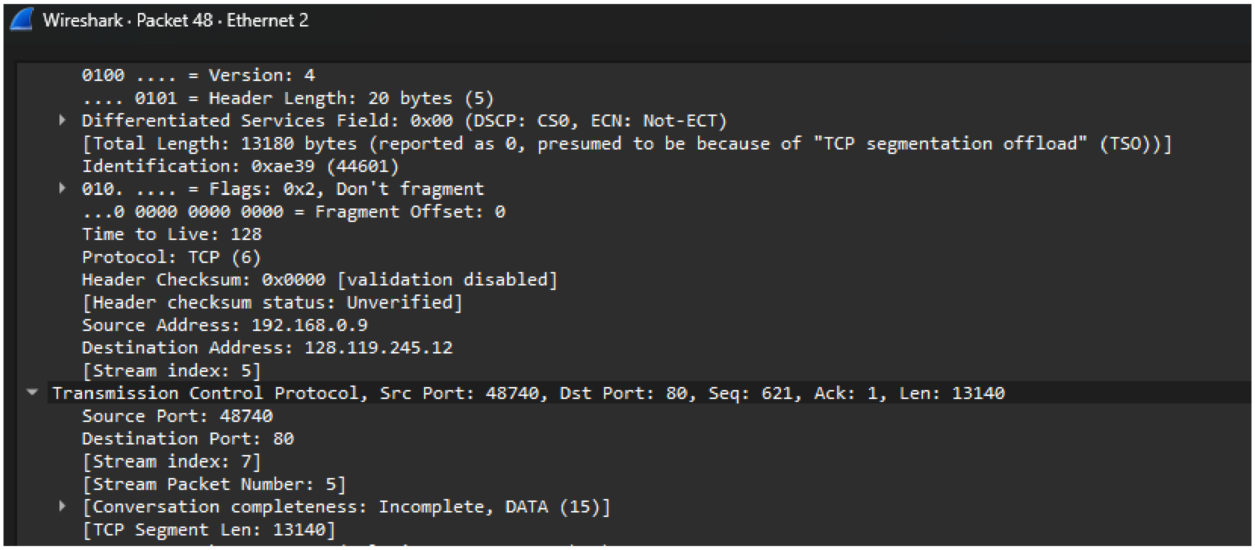
48740

3. What is the IP address of gaia.cs.umass.edu? (**Destination**)

128.119.245.12

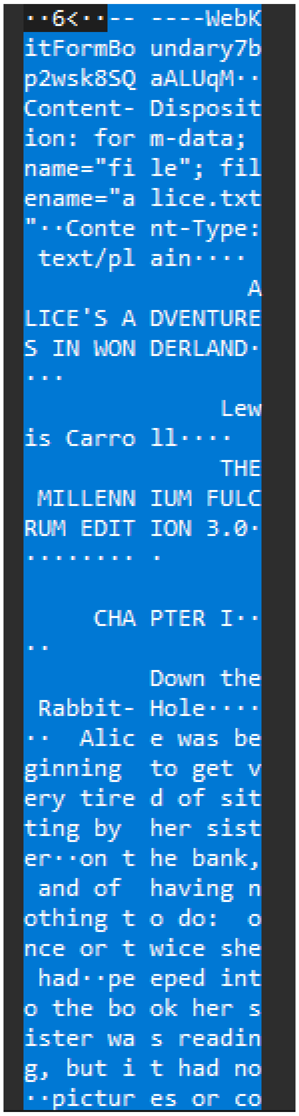
4. On what port number is gaia.cs.umass.edu sending and receiving TCP segments for this connection?

80



```
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      [Total Length: 13180 bytes (reported as 0, presumed to be because of "TCP segmentation offload" (TSO))]
      Identification: 0xae39 (44601)
    ▸ 010. .... = Flags: 0x2, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 128
      Protocol: TCP (6)
      Header Checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.0.9
      Destination Address: 128.119.245.12
      [Stream index: 5]
 ▾ Transmission Control Protocol, Src Port: 48740, Dst Port: 80, Seq: 621, Ack: 1, Len: 13140
      Source Port: 48740
      Destination Port: 80
      [Stream index: 7]
      [Stream Packet Number: 5]
    ▸ [Conversation completeness: Incomplete, DATA (15)]
      [TCP Segment Len: 13140]
```

Additional note: Because this is Http and not Https your File is sent in plain text and you can find the story divided up in the tcp traffic:

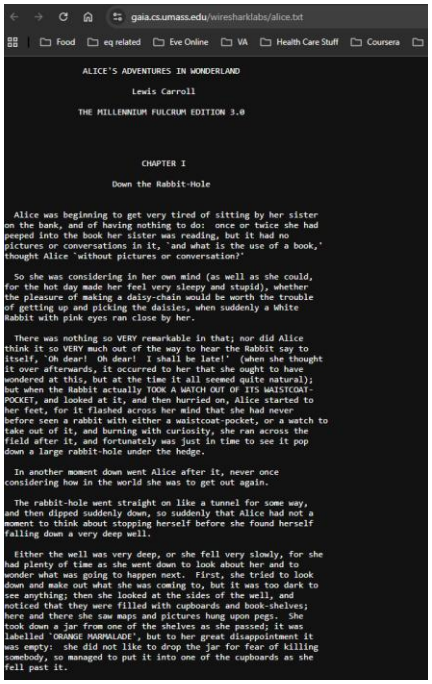


```
··6<··-- ----WebK
itFormBo undary7b
p2wsk8SQ aALUqM··
Content- Disposit
ion: for m-data; 
name="fi le"; fil
ename="a lice.txt
"··Conte nt-Type:
 text/pl ain···· 
                A
LICE'S A DVENTURE
S IN WON DERLAND·
···              
              Lew
is Carro ll····  
              THE
 MILLENN IUM FULC
RUM EDIT ION 3.0·
········ ·       
                 
     CHA PTER I··
··               
         Down the
 Rabbit- Hole····
··  Alic e was be
ginning  to get v
ery tire d of sit
ting by  her sist
er··on t he bank,
 and of  having n
othing t o do:  o
nce or t wice she
 had··pe eped int
o the bo ok her s
ister wa s readin
g, but i t had no
··pictur es or co
```

Summary:

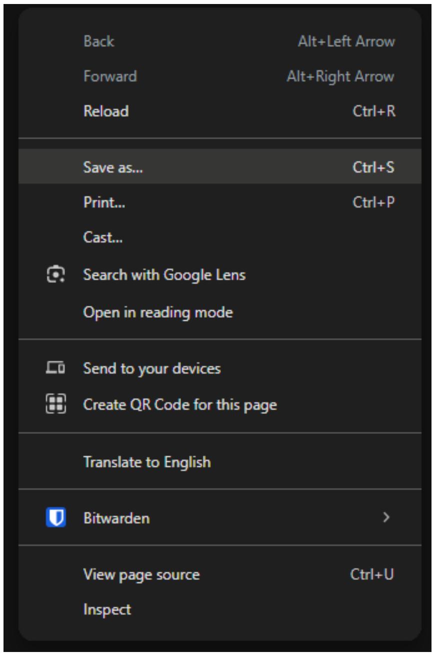
The first step of this lab is to go to the following page:
http://gaia.cs.umass.edu/wiresharklabs/alice.txt

It will look like this:



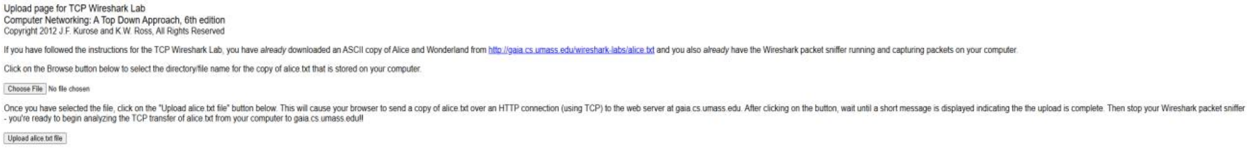
```
              ALICE'S ADVENTURES IN WONDERLAND

                        Lewis Carroll

             THE MILLENNIUM FULCRUM EDITION 3.0




                          CHAPTER I

                     Down the Rabbit-Hole


  Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do:  once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, `and what is the use of a book,'
thought Alice `without pictures or conversation?'

  So she was considering in her own mind (as well as she could,
for the hot day made her feel very sleepy and stupid), whether
the pleasure of making a daisy-chain would be worth the trouble
of getting up and picking the daisies, when suddenly a White
Rabbit with pink eyes ran close by her.

  There was nothing so VERY remarkable in that; nor did Alice
think it so VERY much out of the way to hear the Rabbit say to
itself, `Oh dear!  Oh dear!  I shall be late!'  (when she thought
it over afterwards, it occurred to her that she ought to have
wondered at this, but at the time it all seemed quite natural);
but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-
POCKET, and looked at it, and then hurried on, Alice started to
her feet, for it flashed across her mind that she had never
before seen a rabbit with either a waistcoat-pocket, or a watch to
take out of it, and burning with curiosity, she ran across the
field after it, and fortunately was just in time to see it pop
down a large rabbit-hole under the hedge.

  In another moment down went Alice after it, never once
considering how in the world she was to get out again.

  The rabbit-hole went straight on like a tunnel for some way,
and then dipped suddenly down, so suddenly that Alice had not a
moment to think about stopping herself before she found herself
falling down a very deep well.

  Either the well was very deep, or she fell very slowly, for she
had plenty of time as she went down to look about her and to
wonder what was going to happen next.  First, she tried to look
down and make out what she was coming to, but it was too dark to
see anything; then she looked at the sides of the well, and
noticed that they were filled with cupboards and book-shelves;
here and there she saw maps and pictures hung upon pegs.  She
took down a jar from one of the shelves as she passed; it was
labelled `ORANGE MARMALADE', but to her great disappointment it
was empty:  she did not like to drop the jar for fear of killing
somebody, so managed to put it into one of the cupboards as she
fell past it.
```

You then want to save it to a text file. You can do this by right clicking and Clicking save as Like below **<u>(Take note of where you are saving the file as you will need it later)</u>**:



Once you save the file you will want to navigate to the following page: http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html

The page looks like this:



Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded an ASCII copy of Alice and Wonderland from http://gaia.cs.umass.edu/wireshark-labs/alice.txt and you also *already* have the Wireshark packet sniffer running and capturing packets on your computer.

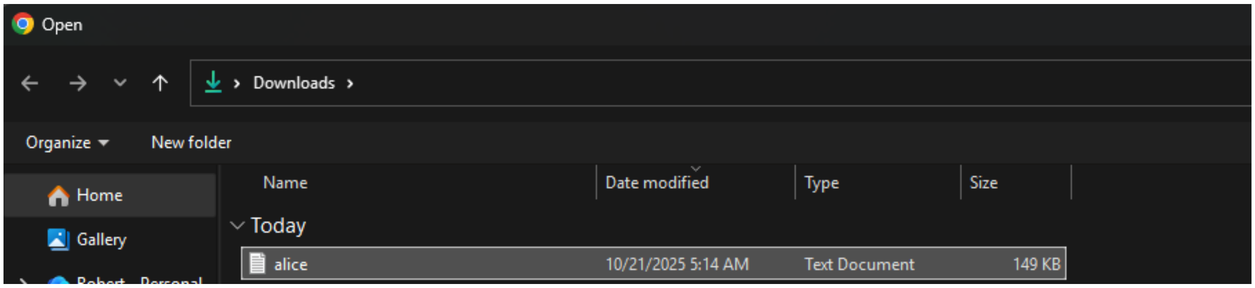
Click on the Browse button below to select the directory/file name for the copy of alice.txt that is stored on your computer.

Choose File No file chosen

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!

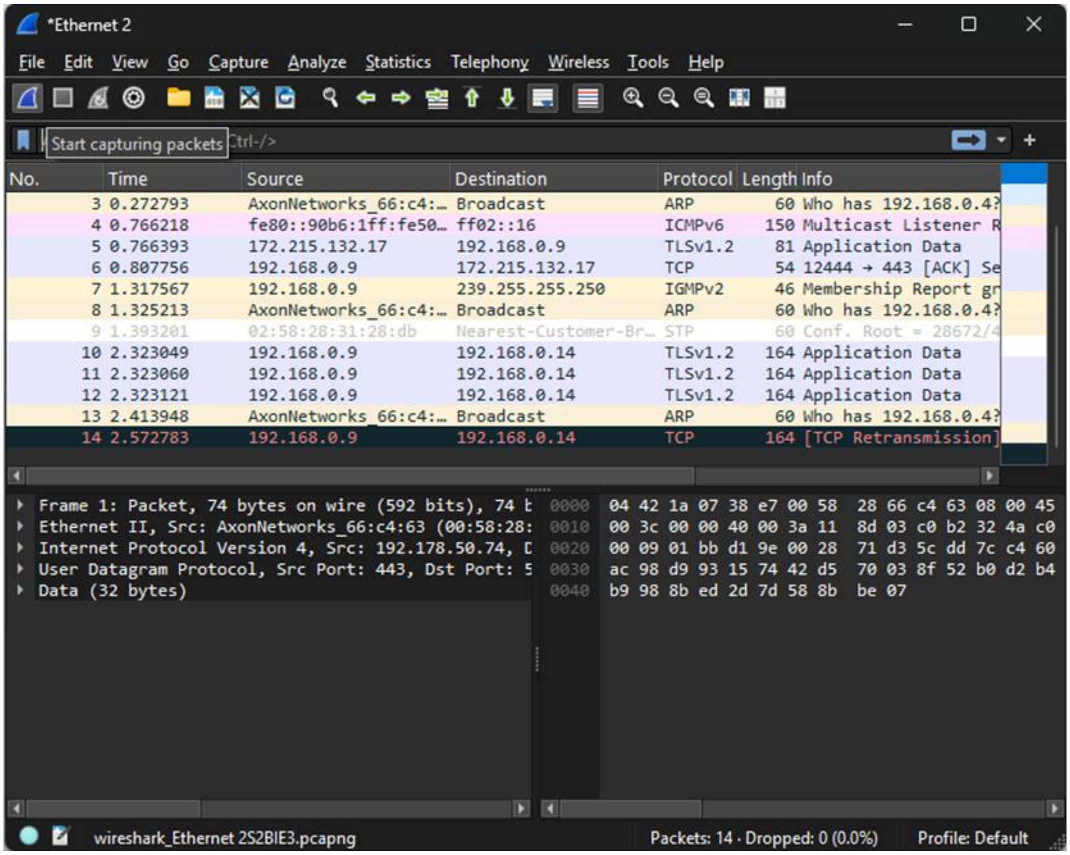
Upload alice.txt file

Next, on the page click Browse or Choose File button a window will appear, select the “alice.txt” document you downloaded in the previous step the window looks like this:



Name	Date modified	Type	Size
Today			
alice	10/21/2025 5:14 AM	Text Document	149 KB

Once your ready start Wireshark and begin capturing traffic, you can do this by clicking the shark fin or by selecting the apator you want to capture traffic from. See below:

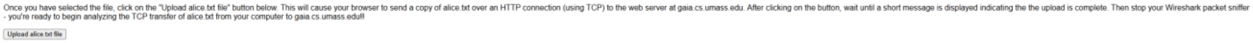


```
No.   Time       Source                    Destination              Protocol Length Info
    3 0.272793   AxonNetworks_66:c4:…      Broadcast                ARP        60 Who has 192.168.0.4?
    4 0.766218   fe80::90b6:1ff:fe50…      ff02::16                 ICMPv6    150 Multicast Listener R
    5 0.766393   172.215.132.17            192.168.0.9              TLSv1.2    81 Application Data
    6 0.807756   192.168.0.9               172.215.132.17           TCP        54 12444 → 443 [ACK] Se
    7 1.317567   192.168.0.9               239.255.255.250          IGMPv2     46 Membership Report gr
    8 1.325213   AxonNetworks_66:c4:…      Broadcast                ARP        60 Who has 192.168.0.4?
    9 1.393201   02:58:28:31:28:db         Nearest-Customer-Br…     STP        60 Conf. Root = 28672/4
   10 2.323049   192.168.0.9               192.168.0.14             TLSv1.2   164 Application Data
   11 2.323060   192.168.0.9               192.168.0.14             TLSv1.2   164 Application Data
   12 2.323121   192.168.0.9               192.168.0.14             TLSv1.2   164 Application Data
   13 2.413948   AxonNetworks_66:c4:…      Broadcast                ARP        60 Who has 192.168.0.4?
   14 2.572783   192.168.0.9               192.168.0.14             TCP       164 [TCP Retransmission]
```

```
› Frame 1: Packet, 74 bytes on wire (592 bits), 74 b       0000  04 42 1a 07 38 e7 00 58  28 66 c4 63 08 00 45
› Ethernet II, Src: AxonNetworks_66:c4:63 (00:58:28:       0010  00 3c 00 00 40 00 3a 11  8d 03 c0 b2 32 4a c0
› Internet Protocol Version 4, Src: 192.178.50.74, D       0020  00 09 01 bb d1 9e 00 28  71 d3 5c dd 7c c4 60
› User Datagram Protocol, Src Port: 443, Dst Port: 5       0030  ac 98 d9 93 15 74 42 d5  70 03 8f 52 b0 d2 b4
› Data (32 bytes)                                          0040  b9 98 8b ed 2d 7d 58 8b  be 07
```

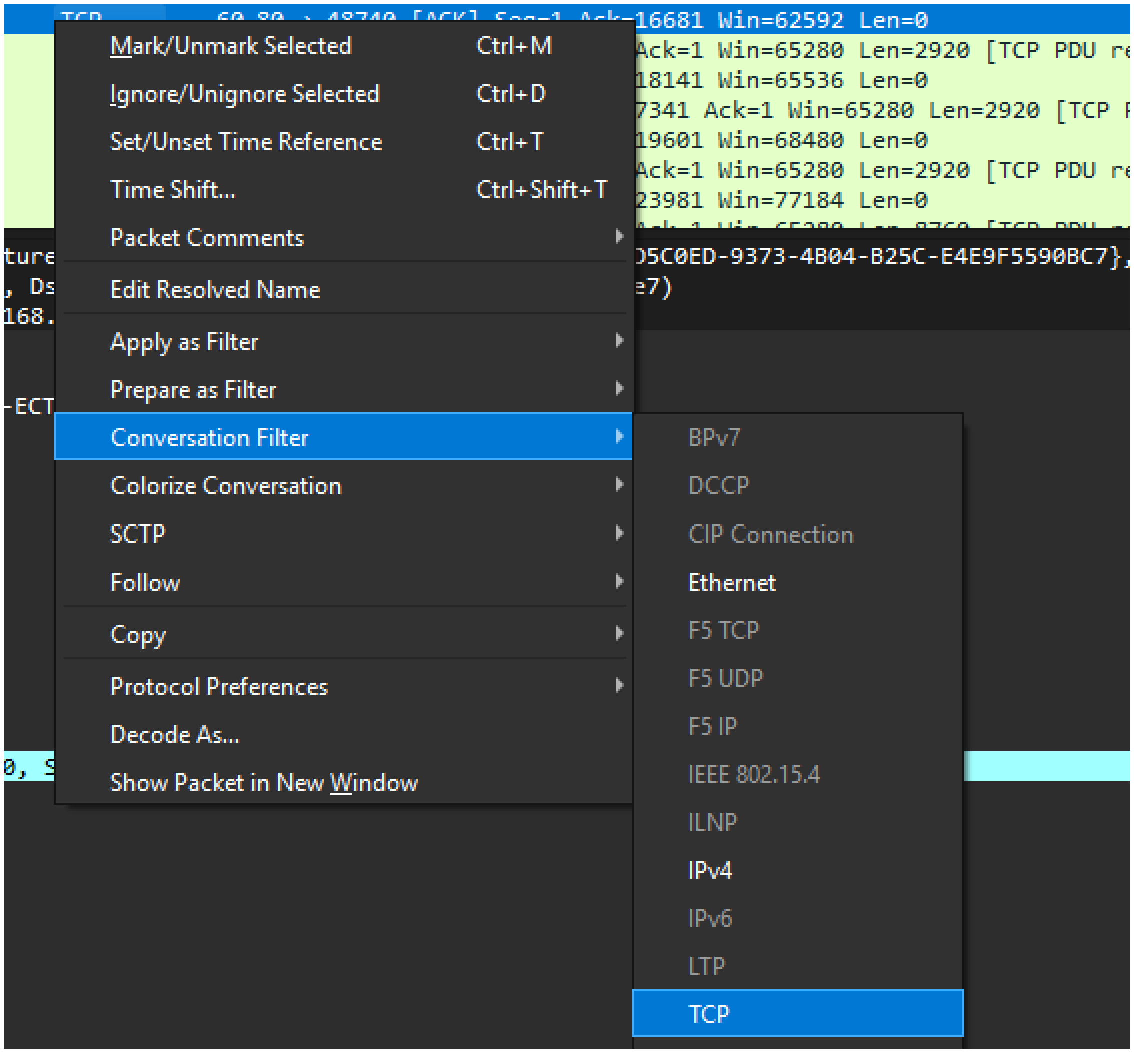
```
wireshark_Ethernet 2S2BIE3.pcapng          Packets: 14 · Dropped: 0 (0.0%)     Profile: Default
```

With Wireshark running, return to the web browser and click the Upload alice.txt button seen below:



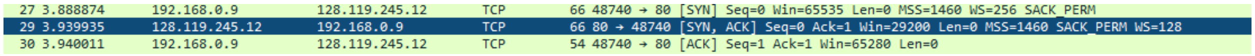
Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!

You should get a Success message from the Website. After this message is received. Return to Wireshark and Stop the packet capture. Find the packets that are heading out with the message in plain text and right click on and click “Filter Conversation with the TCP option” See below:



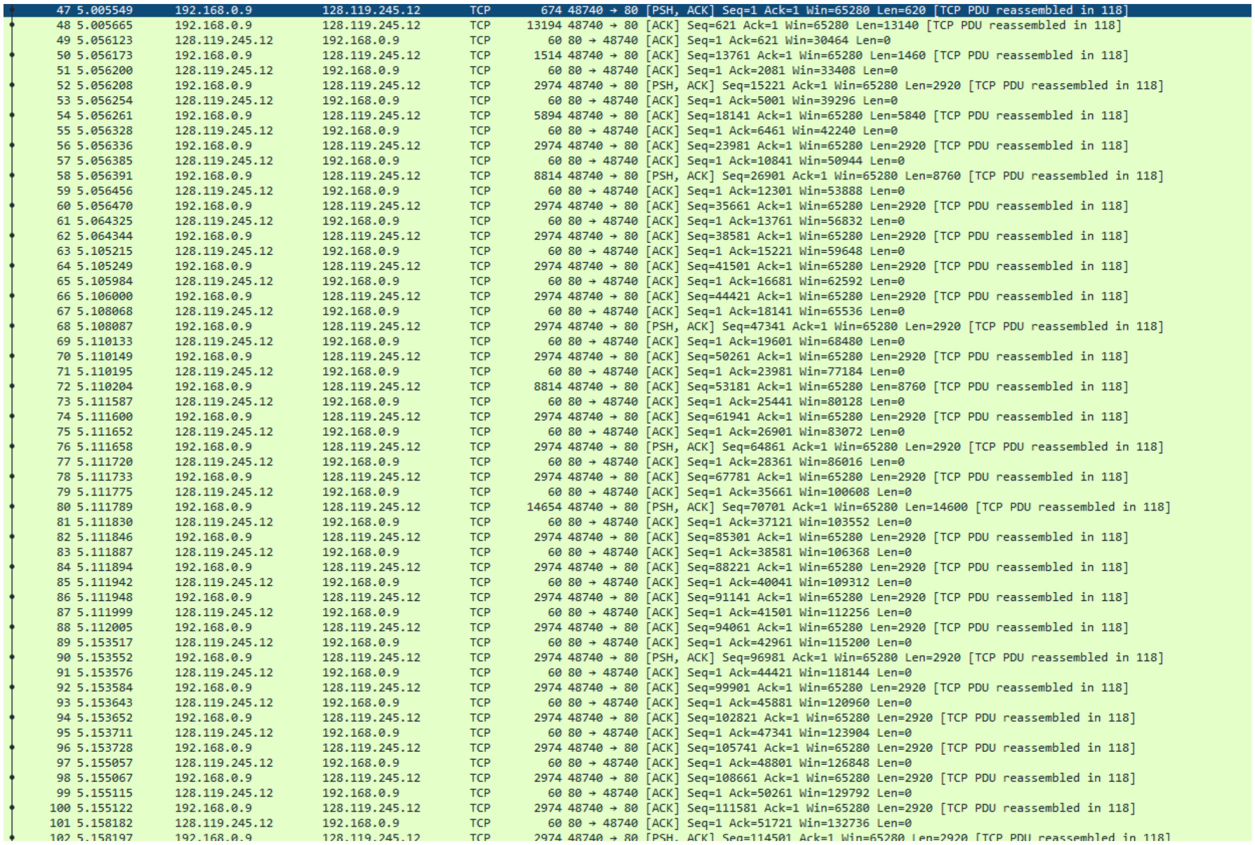
```
TCP   60 80 → 48740 [ACK] Seq=1 Ack=16681 Win=62592 Len=0
Ack=1 Win=65280 Len=2920 [TCP PDU re
18141 Win=65536 Len=0
7341 Ack=1 Win=65280 Len=2920 [TCP P
19601 Win=68480 Len=0
Ack=1 Win=65280 Len=2920 [TCP PDU re
23981 Win=77184 Len=0
ture
, Ds
168.
D5C0ED-9373-4B04-B25C-E4E9F5590BC7},
e7)
-ECT
0, S
```

Here we can See the Originating Syn, Syn,Ack , Ack:



```
 27 3.888874   192.168.0.9       128.119.245.12   TCP   66 48740 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
 29 3.939935   128.119.245.12    192.168.0.9      TCP   66 80 → 48740 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
 30 3.940011   192.168.0.9       128.119.245.12   TCP   54 48740 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
```

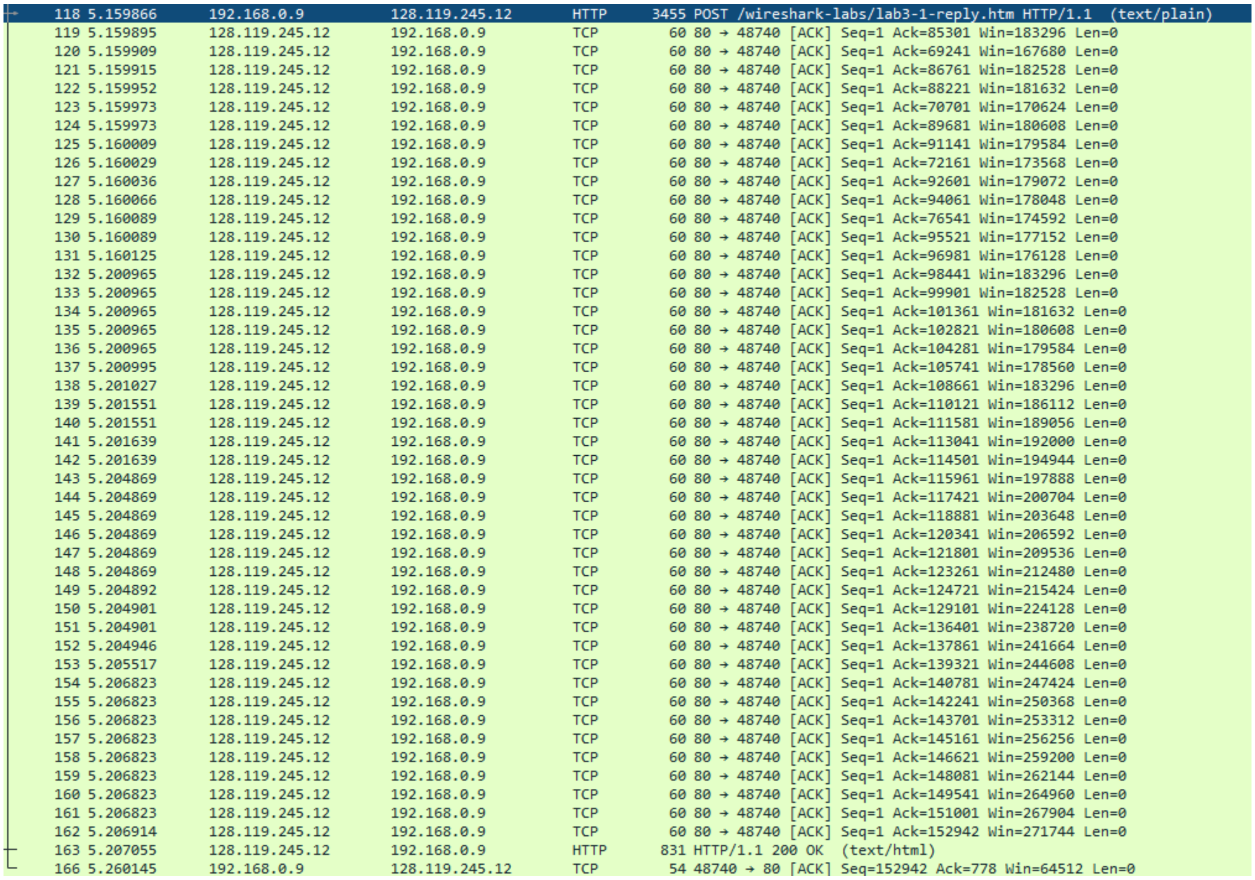
The upload traffic should look like this:



```
 47 5.005549   192.168.0.9       128.119.245.12    TCP    674 48740 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=620 [TCP PDU reassembled in 118]
 48 5.005665   192.168.0.9       128.119.245.12    TCP  13194 48740 → 80 [ACK] Seq=621 Ack=1 Win=65280 Len=13140 [TCP PDU reassembled in 118]
 49 5.056123   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=621 Win=30464 Len=0
 50 5.056173   192.168.0.9       128.119.245.12    TCP   1514 48740 → 80 [ACK] Seq=13761 Ack=1 Win=65280 Len=1460 [TCP PDU reassembled in 118]
 51 5.056200   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=2081 Win=33408 Len=0
 52 5.056208   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [PSH, ACK] Seq=15221 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 53 5.056254   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=5001 Win=39296 Len=0
 54 5.056261   192.168.0.9       128.119.245.12    TCP   5894 48740 → 80 [ACK] Seq=18141 Ack=1 Win=65280 Len=5840 [TCP PDU reassembled in 118]
 55 5.056328   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=6461 Win=42240 Len=0
 56 5.056336   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=23981 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 57 5.056385   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=10841 Win=50944 Len=0
 58 5.056391   192.168.0.9       128.119.245.12    TCP   8814 48740 → 80 [PSH, ACK] Seq=26901 Ack=1 Win=65280 Len=8760 [TCP PDU reassembled in 118]
 59 5.056456   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=12301 Win=53888 Len=0
 60 5.056470   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=35661 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 61 5.064325   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=13761 Win=56832 Len=0
 62 5.064344   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=38581 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 63 5.105215   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=15221 Win=59648 Len=0
 64 5.105249   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=41501 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 65 5.105984   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=16681 Win=62592 Len=0
 66 5.106000   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=44421 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 67 5.108068   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=18141 Win=65536 Len=0
 68 5.108087   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [PSH, ACK] Seq=47341 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 69 5.110133   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=19601 Win=68480 Len=0
 70 5.110149   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=50261 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 71 5.110195   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=23981 Win=77184 Len=0
 72 5.110204   192.168.0.9       128.119.245.12    TCP   8814 48740 → 80 [ACK] Seq=53181 Ack=1 Win=65280 Len=8760 [TCP PDU reassembled in 118]
 73 5.111587   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=25441 Win=80128 Len=0
 74 5.111600   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=61941 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 75 5.111652   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=26901 Win=83072 Len=0
 76 5.111658   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [PSH, ACK] Seq=64861 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 77 5.111720   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=28361 Win=86016 Len=0
 78 5.111733   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=67781 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 79 5.111775   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=35661 Win=100608 Len=0
 80 5.111789   192.168.0.9       128.119.245.12    TCP  14654 48740 → 80 [PSH, ACK] Seq=70701 Ack=1 Win=65280 Len=14600 [TCP PDU reassembled in 118]
 81 5.111830   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=37121 Win=103552 Len=0
 82 5.111846   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=85301 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 83 5.111887   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=38581 Win=106368 Len=0
 84 5.111894   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=88221 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 85 5.111942   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=40041 Win=109312 Len=0
 86 5.111948   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=91141 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 87 5.111999   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=41501 Win=112256 Len=0
 88 5.112005   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=94061 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 89 5.153517   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=42961 Win=115200 Len=0
 90 5.153552   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [PSH, ACK] Seq=96981 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 91 5.153576   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=44421 Win=118144 Len=0
 92 5.153584   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=99901 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 93 5.153643   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=45881 Win=120960 Len=0
 94 5.153652   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=102821 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 95 5.153711   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=47341 Win=123904 Len=0
 96 5.153728   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=105741 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 97 5.155057   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=48801 Win=126848 Len=0
 98 5.155067   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=108661 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
 99 5.155115   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=50261 Win=129792 Len=0
100 5.155122   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [ACK] Seq=111581 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
101 5.158182   128.119.245.12    192.168.0.9       TCP     60 80 → 48740 [ACK] Seq=1 Ack=51721 Win=132736 Len=0
102 5.158197   192.168.0.9       128.119.245.12    TCP   2974 48740 → 80 [PSH, ACK] Seq=114501 Ack=1 Win=65280 Len=2920 [TCP PDU reassembled in 118]
```

You will then see a bunch of TCP traffic ending with a Http Post message. This is the connection acknowling the end of the upload and sending you to the congratulations page.

It should look like this:



```
118 5.159866   192.168.0.9      128.119.245.12   HTTP   3455 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
119 5.159895   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=85301 Win=183296 Len=0
120 5.159909   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=69241 Win=167680 Len=0
121 5.159915   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=86761 Win=182528 Len=0
122 5.159952   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=88221 Win=181632 Len=0
123 5.159973   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=70701 Win=170624 Len=0
124 5.159973   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=89681 Win=180608 Len=0
125 5.160009   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=91141 Win=179584 Len=0
126 5.160029   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=72161 Win=173568 Len=0
127 5.160036   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=92601 Win=179072 Len=0
128 5.160066   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=94061 Win=178048 Len=0
129 5.160089   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=76541 Win=174592 Len=0
130 5.160089   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=95521 Win=177152 Len=0
131 5.160125   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=96981 Win=176128 Len=0
132 5.200965   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=98441 Win=183296 Len=0
133 5.200965   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=99901 Win=182528 Len=0
134 5.200965   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=101361 Win=181632 Len=0
135 5.200965   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=102821 Win=180608 Len=0
136 5.200965   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=104281 Win=179584 Len=0
137 5.200995   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=105741 Win=178560 Len=0
138 5.201027   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=108661 Win=183296 Len=0
139 5.201551   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=110121 Win=186112 Len=0
140 5.201551   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=111581 Win=189056 Len=0
141 5.201639   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=113041 Win=192000 Len=0
142 5.201639   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=114501 Win=194944 Len=0
143 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=115961 Win=197888 Len=0
144 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=117421 Win=200704 Len=0
145 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=118881 Win=203648 Len=0
146 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=120341 Win=206592 Len=0
147 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=121801 Win=209536 Len=0
148 5.204869   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=123261 Win=212480 Len=0
149 5.204892   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=124721 Win=215424 Len=0
150 5.204901   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=129101 Win=224128 Len=0
151 5.204901   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=136401 Win=238720 Len=0
152 5.204946   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=137861 Win=241664 Len=0
153 5.205517   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=139321 Win=244608 Len=0
154 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=140781 Win=247424 Len=0
155 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=142241 Win=250368 Len=0
156 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=143701 Win=253312 Len=0
157 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=145161 Win=256256 Len=0
158 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=146621 Win=259200 Len=0
159 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=148081 Win=262144 Len=0
160 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=149541 Win=264960 Len=0
161 5.206823   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=151001 Win=267904 Len=0
162 5.206914   128.119.245.12   192.168.0.9      TCP      60 80 → 48740 [ACK] Seq=1 Ack=152942 Win=271744 Len=0
163 5.207055   128.119.245.12   192.168.0.9      HTTP    831 HTTP/1.1 200 OK  (text/html)
166 5.260145   192.168.0.9      128.119.245.12   TCP      54 48740 → 80 [ACK] Seq=152942 Ack=778 Win=64512 Len=0
```

You should now be able to easly awnser the questions Below:

1. What is the IP address used by your computer (source)?

2. On what port number is your computer sending and receiving TCP segments for this connection?

3. What is the IP address of gaia.cs.umass.edu? (Destination)

4. On what port number is gaia.cs.umass.edu sending and receiving TCP segments for this connection?